

Instruction Class Diagram -lineNumber : int -parameters : vector<Parameter*> -ptrExecute : InstructionFunction -JMPFlag: bool -microInstructions: vector<MicroInstruction> -ZFlag : bool -NFlag : bool -parseParameters(paramStrings, variables, outFileName) -typeCheckParams(instructionName : string) +Instruction(p_lineNumber, paramStrings, p_ptrExecute, variables, outFileName) MISProgram +execute() fileName: string +getLine Number(): int variables : vector<Parameter*> +getParameters(): vector<Parameter*> * InstructionFunction is a type of function pointer that points to a function instructions : vector<Instruction*> +getMicroInstructions(): vector<MicroInstruction> * that returns void and takes in instruction: Instruction* errFileName : string +getJMPFlag(): bool -outFileName : string +setJMPFlag(flag : bool) variableCloners : map<string, Parameter*> InstructionFunctions holds all of the different functions +getZFlag(): bool instructionFunctionPtrs: map<string, InstructionFunction> that follow the InstructionFunction format +setZFlag(flag : bool) error : bool +getNFlag(): bool _ 1:1 loadLine(lineNumber, line, doneWithVars, JMPs, labels) InstructionFunctions +setNFlag(flag : bool) linkJumps(JMPs, labels) +~Instruction() +add(instruction : Instruction *) +MISProgram(filename : string) +sub(instruction : Instruction *) +clean() +div(instruction : Instruction *) Hoad() +mul(instruction : Instruction *) +run() +assign(instruction : Instruction *) +errorHappened(): bool +out(instruction : Instruction *) ~MISProgram() +setStrChar(instruction : Instruction *) +getStrChar(instruction : Instruction *) +sleep(instruction : Instruction *) Parameter +jmp(instruction : Instruction *) +jmpZ(instruction : Instruction *) type : char Variable<T> +jmpNZ(instruction : Instruction *) isConstant : bool +jmpGT(instruction : Instruction *) value: T name: string +jmpLT(instruction : Instruction *) +Variable() Parameter() +jmpGTE(instruction : Instruction *) +Variable(p_type, name, p_isConstant, p_value) getValue(): T & +jmpLTE(instruction : Instruction *) +getValue(): T & setValue(value: T) +setValue(value: T) +getName(): string & +initialize(paramStrings : vector<string> &) +getType(): char MicroInstructions holds all of the different functions +clone(paramStrings : vector<string> &) : Parameter * +getIsConstant(): bool that follow the MicroInstruction format +~Variable() +initialize (paramStrings: vector<string> &) +clone(paramStrings: vector<string> &): Parameter* MicroInstructions ~Parameter() +ADDR(instruction: Instruction*) +ADDN(instruction : Instruction *) +ADDAR(instruction: Instruction*) +ADDAN(instruction: Instruction*) +ADDAF(instruction : Instruction *) +SUBR(instruction : Instruction *) +SUBN(instruction : Instruction *) +MULR(instruction : Instruction *) +MULN(instruction: Instruction*) +MULAR(instruction: Instruction*) +MULAN(instruction: Instruction*) +DIVR(instruction : Instruction *) +DIVN(instruction : Instruction *) Powered By Visual Paradigm Community Edition &

User to Program
Sequence Diagram





